

**NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
CONTRACT NO. NAS 7-918**

TECHNICAL SUPPORT PACKAGE

On

COUPLED-LAYER ARCHITECTURE FOR ADVANCED SOFTWARE FOR ROBOTS

for 12/01/2002

NASA TECH BRIEF Vol. 26, No. 12

from

JPL NEW TECHNOLOGY REPORT NPO- 21218

Inventor(s):

Darren H Mutz

Hari Das

Richard Petras

Issa A Nesnas

Richard Volpe

Tara A Estlin

NOTICE

This document was prepared under the sponsorship of the National Aeronautics and Space Administration. Neither the United States Government nor any person acting on behalf of the United States Government assumes any liability resulting from the use of the information contained in this document or warrants that such use will be free from privately owned rights. If trade names or manufacturers' names are used in this report, it is for identification only. This usage does not constitute an official endorsement, either expressed or implied, by the National Aeronautics and Space Administration.

TSP assembled by:

JPL Intellectual Assets office

pp. i-ii, 1-3

**JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA**

12/01/2002

Coupled-Layer Architecture for Advanced Software for Robots

Decision-making and functional infrastructures interact at all levels of granularity.

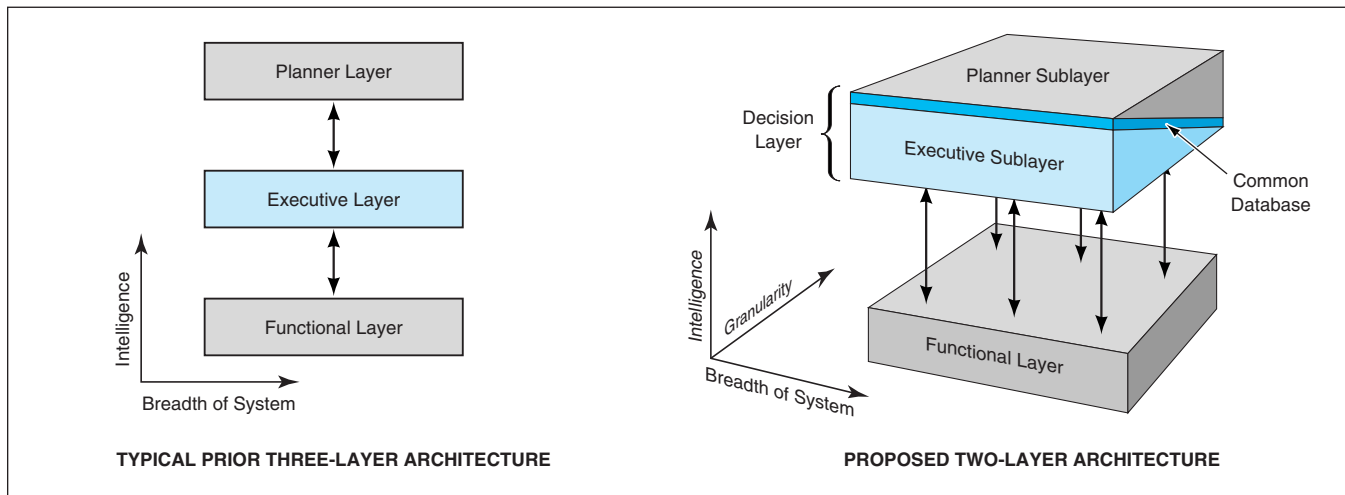
NASA's Jet Propulsion Laboratory, Pasadena, California

The title "Coupled Layer Architecture for Robotics Autonomy" (CLARATy) refers to a software architecture for robots that has been proposed to (1) improve the modularity of robotic-system software while (2) tightening the coupling between autonomy and control software subsystems. Whereas prior robotic architectures have typically been characterized by three layers, the CLARATy is characterized by only two layers. The CLARATy provides for interaction of decision-making and functional infrastructures at all levels of system granularity. This architecture is flexible enough to encompass research and application domains, and provides

for an explicit coupling of artificial-intelligence and robotics techniques. The architecture is also implemented in an object-oriented fashion that makes it possible to leverage software design through both inheritance and aggregation, thereby eliminating the need for duplication of effort in the development of new software.

In a typical three-layer architecture (see figure), the dimension along each layer can be thought of as the breadth of the system in terms of hardware and capabilities. The dimension up from one layer to the next can be thought of as increasing intelligence, from reflexive, to procedural, to deliberative. However, the responsibilities and

height of each layer are not strictly defined, and the line between the planner and executive layers can be blurred. Another shortcoming of a typical prior three-layer architecture is lack of access of the planner to the functional layer. While this lack of access is typically desirable during execution, it separates the planner from information on functionality of the system during planning. One consequence is that a planner often carries its own model of the robotic system, which model may not be directly derived from the model carried in the functional layer. This not only entails repetition of information storage but also often leads to inconsistencies. Still another shortcoming of a typical three-layer architecture is



The **Decision Layer** in the **CLARATy** performs functions that include (among others) those that, in older three-layer architectures, are performed by separate planner and executive layers.

that it misrepresents the granularity in the system and obscures the hierarchies that can exist within the three layers.

In the CLARATy, the planner and executive layers are replaced by a single decision layer that performs both planning and executive functions. The CLARATy offers two major advantages over a typical three-layer architecture: explicit representation of the granularity in a third dimension and blending of the declarative and procedural techniques for decision making. The addition of the granularity dimension enables explicit representation of the hierarchies in the functional layer while accounting for the *de facto* nature of planning horizons in the decision layer. For the functional layer, an object-oriented hierarchy describes the nested encapsulation of subsystems of the system and provides basic capabilities at each level of the nesting. For instance, a command to "move" could be directed at a motor, appendage, mobile robot, or team. For the decision layer, granularity maps to an activities time line that is being created and executed. Because of the nature of the dynamics of the physical system controlled by the functional layer, there is a strong correlation between its system granularity and the time-line granularity of the decision layer.

The blending of declarative and procedural techniques in the decision layer emerges from the trend of planning and scheduling systems that have executive qualities and vice versa. This trend has been a cumulative result of recent advances in algorithms for robotic systems and the development of computers capable of processing data at higher speeds. The CLARATy enhances this trend by explicitly providing for access to the functional layer at higher levels of granularity (that is, larger grains), and hence less frequently, thereby providing more time

for iterative replanning. However, it is still recognized that there is a need for procedural system capabilities, both in (1) the interface between the executive and functional layers and (2) the infusion of procedural semantics for specifying plans and scheduling operations. Therefore, the CLARATy includes a single database at the interface between planning and executive sublayers, leveraging recent efforts to merge the executive and planner layers.

This work was done by Darren Mutz, Hari Das, Issa Nesnas, Richard Petras, Richard Volpe, and Tara Estlin of Caltech for NASA's Jet Propulsion Laboratory. For further information, access the Technical Support Package (TSP) free on-line at www.nasatech.com/tsp under the Information Sciences category.

This software is available for commercial licensing. Please contact Don Hart of the California Institute of Technology at (818) 393-3425. Refer to NPO-21218.

The Most Popular PCI and PC/104 Data Acquisition Boards

Products from
ACCES I/O

ANALOG & DIGITAL I/O BOARDS
BUS EXPANSION KITS
SERIAL COMMUNICATIONS
WATCHDOG TIMERS
PCI BOARDS
DISTRIBUTED I/O
PC/104

ACCES I/O PRODUCTS, INC.
10623 Roselle St., San Diego, CA 92121
Fax: 858-550-7322
email: sales@accessio.com
800-326-1649
www.thisad.info/ACCES-2-1

- Digital I/O from 24 to 120 bits including models with change of state detection!
- 8 and 16-channel digital input/relay output boards
- Isolated serial communications
- 8-port RS485 and RS422
- Configurable A/D and D/A converter boards

NTR INVENTOR'S REPORT
NTR: 21218

**PLEASE BE AS CLEAR AND SPECIFIC AS POSSIBLE, AS THIS REPORT MAY BE
MADE AVAILABLE THROUGH TECH BRIEFS**

Section 1 (Novelty), 2A (Problem), and 2B (Solution) must be completely fully. Your published paper may be attached to satisfy Section 2C (Description and Explanation).

1. Novelty- Describe what is new and different about your work and its improvements over the prior art. Attach supporting material if necessary.

Unlike previous robotic architectures, we propose a novel two layer design, that allows decision-making and functional infrastructure to interact at all levels of system granularity. This design is flexible enough to encompass research and application domains, and provides for an explicit coupling of Artificial Intelligence and Robotics techniques. The design is also implemented in an object-oriented fashion, enabling leveraging of software design through both inheritance and aggregation, thereby eliminating duplication of effort seen in previous designs.

2. Technical Disclosure

A. Problem-Motivation that led to development or problem that was solved.

The development of Robotics and Autonomy architecture is as old as the field itself. Therefore, it is not possible here to completely review the body of work upon which this effort builds. Instead, we will simply describe some of the more recent or dominant trends influencing the new architecture presented in this document.

Efforts in robotic architectures have largely arisen from a pragmatic need to structure the software development for ease of system building. As such, they have grown in scope and complexity as the corresponding systems have grown. Early efforts concentrated in detailed software packages~\cite{hayward86}, or general frameworks~\cite{albus87}. Only in the last decade, with the emergence of fast computers with real-time operating systems, have infrastructures been designed as open-architecture controllers of modern robot systems~\cite{volpe97c,schneider98,borrelly98}.

In parallel with robot control efforts, artificial intelligence systems for planning/scheduling and execution were developed which relied on underlying closed-architecture robot controllers~\cite{firby89,simmons94}. The tendency of these systems to be slow and computationally costly led to the emergence of a minimalist school of thought using Behavior Control~\cite{brooks86}. But with faster control layers available, and a general desire to leverage planning functionality, newer systems implement a multi-tiered approach that includes planning, execution, and control in one modern software framework~\cite{alami98,albus00}.

While these end-to-end architectures have been prototyped, some problems have emerged. First, there is no generally accepted standard, preventing leverage of the entire community's effort. This problem has led to the second, which is that implemented systems have typically emerged as a patchwork of legacy and other code not designed to work together. Third, robotics implementations have been slow to leverage the larger industry standards for object-oriented software development, within the Unified Modeling Language (UML) framework. Therefore, we believe the time is ripe to revisit robotics and autonomy efforts with fresh effort aimed at addressing these shortcomings.

B. Solution

To correct the shortfalls in the three-level architecture, we propose an evolution to a two-tiered Coupled Layer Autonomous Robot Architecture (CLARAty). This structure has two major advantages: explicit representation of the system layers' granularity as a third dimension, and blending of the declarative and procedural techniques for decision making.

The addition of a granularity dimension allows for explicit representation of the system hierarchies in the Functional Layer, while accounting for the {\sl de facto} nature of planning horizons in the Decision Layer. For the Functional Layer, an object oriented hierarchy describes the system's nested encapsulation of subsystems, and provides basic capabilities at each level of the nesting. For instance, a command to ``move" could be directed at a motor, appendage, mobile robot, or team. For the Decision Layer, granularity maps to the activities time-line being created and executed. Due to the nature of the dynamics of the physical system controlled by the Functional Layer, there is a strong correlation between its system granularity and the time-line granularity of the Decision Layer.

The blending of declarative and procedural techniques in the Decision Layer emerges from the trend of Planning and Scheduling systems that have Executive qualities and vice versa. This has been afforded by algorithmic and system advances, as well as faster processing. CLARAty enhances this trend by explicitly providing for access of the Functional Layer at higher levels of granularity, thus less frequently, allowing more time for iterative replanning. However, it is still recognized that there is a need for procedural system capabilities in both the Executive interface to the Functional Layer, as well as the infusion of procedural semantics for plan specification and scheduling operations. Therefore, CLARAty has a single database to interface Planning and Executive Functionality, leveraging recent efforts to merge these capabilities.

C. Detailed Description and Explanation

Please see publication for more details.

***PUBLICATION**

R. Volpe, I. Nesnas, T. Estlin, D. Mutz, R. Petras, H. Das, "The CLARAty Architecture for Robotic Autonomy." Proceedings of the 2001 IEEE Aerospace Conference, Big Sky, Montana, March 10-17, 2001

R. Volpe, I.A.D. Nesnas, T. Estlin, D. Mutz, R. Petras, H. Das, "CLARAty: Coupled Layer Architecture for Robotic Autonomy." JPL Technical Report D-19975, Dec 2000. pdf (116 pages, 904 KB)

<http://telerobotics.jpl.nasa.gov/tasks/claraty/reports/publications/>

*Please obtain publications from sources listed.

“Reference herein to any specific commercial product, process or service by trade name, trademark manufacturer or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.”

“The work described here was carried out at the Jet Propulsion Laboratory, California Institute of Technology under contract with the National Aeronautics and Space Administration.”